

Finite mixture models in **secr** 5.2

Murray Efford

2025-01-26

Contents

Introduction	1
Implementation in secr	1
Number of classes	3
Multimodality	4
Hybrid ‘hcov’ model	5
Notes	5
References	5
Appendix. SECR finite mixture model in which some individuals are of known class	6

Introduction

Variation in detection probability among individuals (‘individual heterogeneity’) is a persistent problem in capture–recapture studies. Ideally, such variation is modelled by grouping individuals into homogeneous classes (males and females) or including continuous predictors such as body weight. Finite mixture models are an option when unmodelled heterogeneity remains (Pledger 2000; Borchers and Efford 2008). The population is assumed to comprise 2 or more latent classes differing in detection parameters, with an unknown proportion in each class. The likelihood uses a weighted sum over the classes, where the weight(s) (mixing proportion(s)) are parameters to be estimated.

Mixture models are prone to fitting problems caused by multimodality of the likelihood. Some comments are offered below, but a fuller investigation is needed.

The distinction between a finite mixture model and one in which the classes of individuals are known is removed in a hybrid (‘hcov’) model documented here.

Implementation in secr

secr allows 2- or 3-class finite mixture models for any ‘real’ detection parameter (e.g., g_0 or sigma of a halfnormal detection function). Consider a simple example in which we specify a 2-class mixture by adding the predictor ‘h2’ to the model formula:

```
library(secr)
setNumThreads(18)
```

```
## [1] 18
```

```
options(digits = 6, width = 85)
```

```
captdata <- reduce(captdata, output = 'multi') # to get rid of distracting warning
model.0 <- secr.fit(captdata, model = g0~1, buffer = 100, trace = FALSE)
model.h2 <- secr.fit(captdata, model = g0~h2, buffer = 100, trace = FALSE)
```

```
model.h2
```

```
##
## secr.fit(capthist = captdata, model = g0 ~ h2, buffer = 100,
##   trace = FALSE)
## secr 5.2.0, 12:03:03 26 Jan 2025
##
## Detector type      multi
## Detector number   100
## Average spacing    30 m
## x-range            365 635 m
## y-range            365 635 m
##
## N animals         : 76
## N detections       : 235
## N occasions        : 5
## Mask area          : 21.2271 ha
##
## Model              : D~1 g0~h2 sigma~1 pmix~h2
## Fixed (real)       : none
## Detection fn        : halfnormal
## Distribution        : poisson
## N parameters        : 5
## Log likelihood     : -758.747
## AIC                 : 1527.49
## AICc                : 1528.35
##
## Beta parameters (coefficients)
##      beta      SE.beta      lcl      ucl
## D      1.705391 0.1190288  1.47210  1.93868
## g0     -1.631054 1.7761699 -5.11228  1.85017
## g0.h22  0.820156 1.3405485 -1.80727  3.44758
## sigma   3.380752 0.0449141  3.29272  3.46878
## pmix.h22 1.284767 5.0996148 -8.71029 11.27983
##
## Variance-covariance matrix of beta parameters
##      D      g0      g0.h22      sigma      pmix.h22
## D      0.01416785 -0.02529738  0.02025377 -0.00106941  0.0584968
## g0     -0.02529738  3.15477947 -2.33268353  0.00333896 -8.6289291
## g0.h22  0.02025377 -2.33268353  1.79707038 -0.00480818  6.0578410
## sigma  -0.00106941  0.00333896 -0.00480818  0.00201728 -0.0190110
## pmix.h22 0.05849680 -8.62892910  6.05784096 -0.01901095 26.0060713
##
## Fitted (real) parameters evaluated at base levels of covariates
##
## session = 1, h2 = 1
##      link estimate SE.estimate      lcl      ucl
```

```
## D      log  5.503535    0.657406 4.35837e+00  6.949591
## g0     logit 0.163686    0.243145 5.98626e-03  0.864148
## sigma  log 29.392865    1.320821 2.69160e+01 32.097625
## pmix   logit 0.216740    0.865729 1.26249e-05  0.999835
##
## session = 1, h2 = 2
##      link estimate SE.estimate      lcl      ucl
## D      log  5.503535    0.657406 4.35837e+00  6.949591
## g0     logit 0.307699    0.114017 1.34709e-01  0.559259
## sigma  log 29.392865    1.320821 2.69160e+01 32.097625
## pmix   logit 0.783260    0.865729 1.64853e-04  0.999987
```

You can see that `secre.fit` has expanded the model to include an extra ‘real’ parameter ‘pmix’, for the proportions in the respective latent classes. You could specify this yourself as part of the ‘model’ argument, but `secre.fit` knows to add it. The link function for ‘pmix’ defaults to ‘mlogit’ (after the mlogit link in MARK), and any attempt to change the link is ignored.

There are two extra ‘beta’ parameters: `g0.h22`, which is the difference in `g0` between the classes on the link (logit) scale, and `pmix.h22`, which is the proportion in the second class, also on the logit scale. Fitted (real) parameter values are reported separately for each mixture class (`h2 = 1` and `h2 = 2`). An important point is that exactly the same estimate of total density is reported for both mixture classes; the actual abundance of each class is $D \times \text{pmix}$.

We can compare a 2-class finite mixture model to the null (constant) model using AIC:

```
AIC (model.0, model.h2)
```

```
##              model detectfn npar  logLik      AIC      AICc  dAIC
## model.0      D~1 g0~1 sigma~1 halfnormal    3 -759.026 1524.05 1524.38 0.000
## model.h2    D~1 g0~h2 sigma~1 pmix~h2 halfnormal    5 -758.747 1527.49 1528.35 3.443
##              AICwt
## model.0      0.8483
## model.h2     0.1517
```

In this case there is no reason to prefer the mixture model.

More complex models are allowed. For example, one might, somewhat outlandishly, fit a learned response to capture that differs between two latent classes, while also allowing `sigma` to differ between classes:

```
model.h2xbh2s <- secre.fit(captdata, model = list(g0~h2*b, sigma~h2), CL = FALSE)
```

Number of classes

The theory of finite mixture models in capture–recapture (Pledger 2000) allows an indefinite number of classes – 2, 3 or perhaps more. Programmatically, the extension to more classes is obvious (e.g., `h3` for a 3-class mixture). The appropriate number of latent classes may be determined by comparing AIC for the fitted models¹

Looking on the bright side, it is unlikely that you will ever have enough data to support more than 2 classes. For the data in the example above, the 2-class and 3-class models have identical log likelihood to 4 decimal places, while the latter requires 2 extra parameters to be estimated (this is to be expected as the data were simulated from a null model with no heterogeneity).

¹score tests (e.g. McCrea, R. S. and Morgan, B. J. T. (2011) Multistate mark-recapture model selection using score tests *Biometrics* **67**, 234–241) are not appropriate because the models are not nested, at least that’s how it seems to me.

Multimodality

The likelihood of a finite mixture model may have multiple modes (e.g. Brooks et al. 1997, Pledger 2000). The risk is ever-present that the numerical maximization algorithm will get stuck on a local peak, and in this case the estimates are simply wrong. Slight differences in starting values or numerical method may result in wildly different answers. (See Dawson and Efford (2009) and the vignette `secr-sound.pdf` for another example of a multimodal likelihood in SECR).

The problem has not been explored fully for SECR models, and care is needed. Pledger (2000) recommended fitting a model with more classes as a check in the non-spatial case, but this is not proven to work with SECR models. It is desirable to try different starting values. This can be done simply using another model fit. For example²:

```
model.h2.2 <- secr.fit(captdata, model = g0~h2, start = model.0, CL = TRUE, trace = FALSE)
```

A more time consuming, but illuminating, check on a 2-class model is to plot the profile log likelihood for a range of mixture proportions (Brooks et al. 1997). We can use the function `pmixprofileLL` in `secr` 5.2³ to calculate these profile likelihoods. This requires a maximization step for each value of 'pmix'; multiple cores may be used in parallel to speed up the computation. `pmixprofileLL` expects the user to identify the coefficient or 'beta parameter' corresponding to 'pmix' (argument 'pmi'):

```
# run pmixprofileLL for one year of ovenbird data and plot results
pmvals <- seq(0.01,0.99,0.01)
# use a coarse mask to make it faster
mask <- make.mask(traps(ovenCH[[1]]), nx = 32, buffer = 200, type = "trapbuffer")
profileLL <- pmixProfileLL(ovenCH[[1]], model = list(g0~h2, sigma~h2), pmi = 5,
                           CL = TRUE, pmvals = pmvals, mask = mask, trace = FALSE)
```

```
## Warning in (function (capthist, model = list(D ~ 1, g0 ~ 1, sigma ~ 1), : possible
## maximization error: nlm returned code 4. See ?nlm
plot(pmvals, profileLL, xlim = c(0,1),
     xlab = 'Fixed pmix', ylab = 'Profile log-likelihood')
```

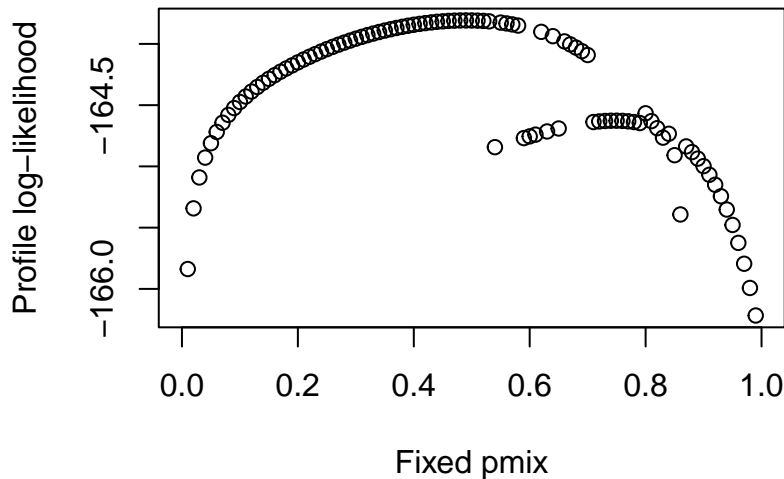


Fig. Profile log-likelihood for mixing proportion between 0.01 and 0.99 in a 2-class finite mixture model (ovenbird data 2005).

Multimodality is likely to show up as multiple rounded peaks in the profile likelihood. Label switching (e.g., Stephens 2000) may cause some ghost reflections about $\text{pmix} = 0.5$ that can be ignored. If multimodality is found one should accept only estimates for which the maximized likelihood matches that from the highest

²Using a null model, as here, actually failed before 2.10 because the start model needed to be nested within the fitted model.

³A slightly different and less general function of the same name was previously included here.

peak. In the ovenbird example, the maximized log likelihood of the fitted h2 model was -163.8 and the estimated mixing proportion was 0.51, so the correct maximum was found.

Maximization algorithms (argument ‘method’ of `secr.fit`) differ in their tendency to settle on local maxima; ‘Nelder-Mead’ is probably better than the default ‘Newton-Raphson’. Simulated annealing is sometimes advocated, but it is slow and has not been tried with SECR models.

Hybrid ‘hcov’ model

The hybrid mixture model accepts a categorical (factor) individual covariate for class membership that may be missing (NA) for any fraction of animals. The name of the covariate to use is specified as argument ‘hcov’ in `secr.fit`. If the covariate is missing for all individuals then a full finite mixture model will be fitted (i.e. mixture as a random effect). Otherwise, the random effect applies only to the animals of unknown class; others are modelled with detection parameter values appropriate to their known class. If class is known for all individuals the model is equivalent to a covariate (CL = TRUE) or grouped (CL = FALSE) model. When many or all animals are of known class the mixing parameter may be treated as an estimate of population proportions (probability a randomly selected individual belongs to class m). This is obviously useful for estimating sex ratio free of detection bias. See the hcov help page (?hcov) for implementation details, and the Appendix for the theory.

Notes

It’s worth mentioning a perennial issue of interpretation: Do the latent classes in a finite mixture model have biological reality? The answer is ‘Probably not’ (although the hybrid model blurs this issue). Fitting a finite mixture model does not require or imply that there is a matching structure in the population (discrete types of animal). A mixture model is merely a convenient way to capture heterogeneity.

When more than one real parameter is modelled as a mixture, there is an ambiguity: is the population split once into latent classes common to all real parameters, or is the population split separately for each real parameter? The second option would require a distinct level of the mixing parameter for each real parameter. `secr` implements only the ‘common classes’ option, which saves one parameter.

References

- Borchers, D. L. and Efford, M. G. (2008) Spatially explicit maximum likelihood methods for capture–recapture studies. *Biometrics* **64**, 377–385.
- Brooks, S. P., Morgan, B. J. T., Ridout, M. S. and Pack, S. E. (1997) Finite mixture models for proportions. *Biometrics* **53**, 1097–1115.
- Dawson, D. K. and Efford, M. G. (2009) Bird population density estimated from acoustic signals. *Journal of Applied Ecology* **46**, 1201–1209.
- Efford, M. G., Borchers D. L. and Byrom, A. E. (2009) Density estimation by spatially explicit capture–recapture: likelihood-based methods. In: D. L. Thomson, E. G. Cooch and M. J. Conroy (eds) *Modeling Demographic Processes in Marked Populations*. Springer. Pp 255–269.
- Efford, M. G., Dawson, D. K. and Borchers, D. L. (2009) Population density estimated from locations of individuals on a passive detector array. *Ecology* **90**, 2676–2682.
- Pledger, S. (2000) Unified maximum likelihood estimates for closed capture–recapture models using mixtures. *Biometrics* **56**, 434–442.
- Stephens, M. (2000) Dealing with label switching in mixture models. *Journal of the Royal Statistical Society, Series B* **62**, 795–809.

Appendix. SECR finite mixture model in which some individuals are of known class

Borchers and Efford (2008 p. 381) gave the likelihood for a Poisson SECR model with U latent classes in proportions $\psi = (\psi_1, \dots, \psi_U)$. For each class u there is an associated vector of detection parameters θ_u (collectively θ). In their notation, n is the number of individuals detected at least once, $\delta. > 0$ refers to animals detected at least once, ω_i is the detection history of animal i , and \mathbf{X} is a point within the potential habitat of the animal (see below for $\Pr\{\omega_i\}$ and $p.(\cdot)$). If the density at point \mathbf{X} is given by $D(\mathbf{X})$, and we omit a constant multinomial term (shown by Borchers and Efford 2008 on p. 378, but dropped without comment on p. 381),

$$L(\theta, \phi, \psi) \propto \frac{\lambda^n e^{-\lambda}}{n!} \prod_{i=1}^n \sum_{u=1}^U \int \frac{\Pr\{\omega_i | X, \theta_u\}}{p.(\mathbf{X}, \theta_u)} f(\mathbf{X}, u | \delta. > 0) d\mathbf{X}$$

where

$$\lambda = \sum_{u=1}^U \psi_u \int D(\mathbf{X}) p.(\mathbf{X}, \theta_u) d\mathbf{X},$$

and

$$f(\mathbf{X}, u | \delta. > 0) = \frac{D(\mathbf{X}) p.(\mathbf{X}, \theta_u) \psi_u}{\sum_{u=1}^U \int D(\mathbf{X}) p.(\mathbf{X}, \theta_u) \psi_u d\mathbf{X}}.$$

Integration is over points within the potential habitat. The probability of each observed history $\Pr\{\omega_i | \mathbf{X}, \theta_u\}$ depends on the detection function and other details of the probability model (Borchers and Efford 2008; Efford, Borchers and Byrom 2009; Efford, Dawson and Borchers 2009), and need not be specified here. The probability of an animal with centre at \mathbf{X} being detected at least once ($p.$) is one minus the probability of observing a null history, computed as for $\Pr(\omega_i)$.

Under this model, class membership is an unobserved random variable. We now modify the likelihood for data in which the class membership of some or all individuals is known. Indicate the class membership of the i -th individual by a variable u_i that may take values $0, 1, \dots, U$, where $u_i = 0$ indicates an individual of unknown class, and the class frequencies are n_0, n_1, \dots, n_U (not to be confused with n_1, \dots, n_C in Borchers and Efford p. 378). We assume detection histories are sorted by class membership, starting with the unknowns.

The Poisson term and the expression for λ remain unchanged, but we must split the product over detected individuals and include a multinomial term for the observed distribution over classes:

$$\begin{aligned} L(\theta, \phi, \psi) &\propto \frac{\lambda^n e^{-\lambda}}{n!} \prod_{i=1}^{n_0} \sum_{u=1}^U \int \frac{\Pr\{\omega_i | X, \theta_u\}}{p.(\mathbf{X}, \theta_u)} f(\mathbf{X}, u | \delta. > 0) d\mathbf{X} \\ &\times \prod_{i=n_0+1}^n \int \frac{\Pr\{\omega_i | X, \theta_{u_i}\}}{p.(\mathbf{X}, \theta_{u_i})} f'(\mathbf{X} | \delta. > 0; u_i) d\mathbf{X} \\ &\times \binom{n - n_0}{n_1, \dots, n_U} \prod_{u=1}^U \left[\frac{\lambda_u}{\lambda} \right]^{n_u}, \end{aligned}$$

where $\lambda_u = \psi_u \int D(\mathbf{X}) p.(\mathbf{X}, \theta_u) d\mathbf{X}$, and the multinomial coefficient $\binom{n - n_0}{n_1, \dots, n_U}$ is a constant that can be omitted. Rather than representing the joint probability density of \mathbf{X} and u_i as in $f(\cdot)$ previously, $f'(\cdot)$ is the probability density of \mathbf{X} for given u_i :

$$f'(\mathbf{X} | \delta. > 0; u_i) = \frac{D(\mathbf{X}) p.(\mathbf{X}, \theta_{u_i})}{\int D(\mathbf{X}) p.(\mathbf{X}, \theta_{u_i}) d\mathbf{X}}.$$

The likelihood conditions on the number of known-class animals detected ($n - n_0$), rather than modelling class identification as a random process. It assumes that the probability that class will be recorded does not depend on class, and that such recording when it happens is without error.

For homogeneous density D the likelihood simplifies to

$$L(\theta, \phi, \psi) \propto \frac{\lambda^n e^{-\lambda}}{n!} \prod_{i=1}^{n_0} \sum_{u=1}^U \int \frac{\Pr\{\omega_i | X, \theta_u\} \psi_u}{\sum_u a(\theta_u) \psi_u} d\mathbf{X} \\ \times \prod_{i=n_0+1}^n \int \frac{\Pr\{\omega_i | X, \theta_{u_i}\}}{a(\theta_{u_i})} d\mathbf{X} \prod_{u=1}^U \left[\frac{a(\theta_u) \psi_u}{\sum_u a(\theta_u) \psi_u} \right]^{n_u},$$

where $a(\theta_u) = \int p(\mathbf{X}, \theta_u) d\mathbf{X}$.

For computation it is convenient to treat the integration as a summation over M small cells, and to compute the log likelihood l . If D is expressed relative to cell area, and we use $\lambda_u = \psi_u \sum_{m=1}^M D(\mathbf{X}_m) p(\mathbf{X}_m, \theta_u)$ and $\lambda = \sum_{u=1}^U \lambda_u$, then

$$l(\theta, \phi, \psi) \propto \log \left(\frac{\lambda^n e^{-\lambda}}{n!} \right) \\ + \sum_{i=1}^{n_0} \log \left(\sum_{u=1}^U \sum_{m=1}^M \Pr\{\omega_i | \mathbf{X}_m, \theta_u\} D(\mathbf{X}_m) \psi_u \right) - n_0 \log(\lambda) \\ + \sum_{i=n_0+1}^n \log \left(\sum_{m=1}^M \Pr\{\omega_i | \mathbf{X}_m, \theta_{u_i}\} D(\mathbf{X}_m) \right) - \sum_{u=1}^U n_u \log(\lambda_u / \psi_u) \\ + \sum_{u=1}^U n_u \log(\lambda_u / \lambda),$$

or, after some cancellation,

$$l(\theta, \phi, \psi) \propto \log \left(\frac{\lambda^n e^{-\lambda}}{n!} \right) - n \log(\lambda) \\ + \sum_{i=1}^{n_0} \log \left(\sum_{u=1}^U \sum_{m=1}^M \Pr\{\omega_i | \mathbf{X}_m, \theta_u\} D(\mathbf{X}_m) \psi_u \right) \\ + \sum_{i=n_0+1}^n \log \left(\sum_{m=1}^M \Pr\{\omega_i | \mathbf{X}_m, \theta_{u_i}\} D(\mathbf{X}_m) \right) + \sum_{u=1}^U n_u \log(\psi_u).$$